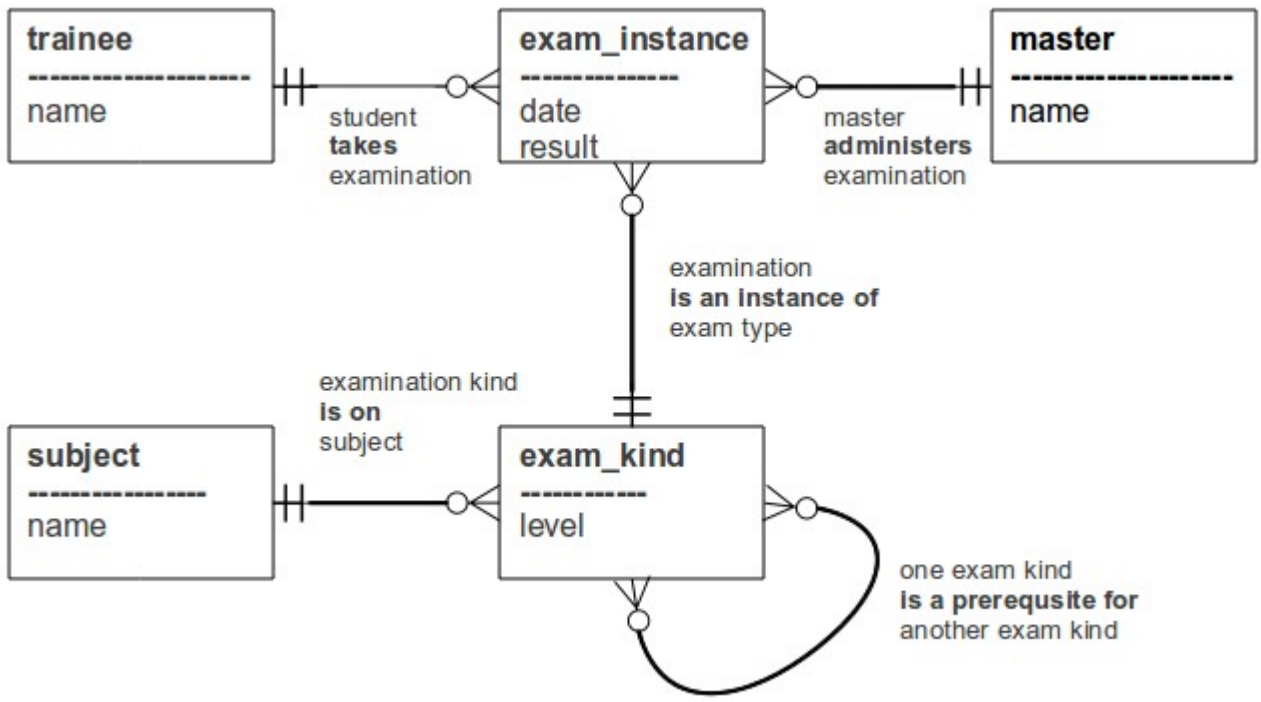
INF1343, Winter 2012, Quiz 2

_____

write your name above

a) Yoda's Jedi Academy is a non-profit organization dedicated to training future Jedi Knights. Yoda wants to build a database to keep track of which of the different standardized Jedi examinations each trainee has taken and whether they passed or failed those examinations. Examples of examinations would include "Light Saber Combat," "Contemporary Jedi History," "Jedi Mind Tricks", or "Linear Algebra." Each exam may be offered at different levels. E.g., the "Light Saber Combat" exam has levels 1, 2, and 3. The examinations are done one-on-one: one trainee is examined by one Jedi Master. (Yoda doesn't have time to administer all the examinations himself.) When a trainee takes an exam, she or he can either pass or fail it. Yoda wants to record the result in either case. A trainee can fail each exam up to three times before they are kicked out of the Academy. Also, in some cases a trainee must successfully pass certain exams before they are allowed to take some of the other exams. For example, one must pass Light Saber Combat Level 2 and Contemporary Jedi History Level 3 before being allowed to take Jedi Mind Tricks Level 1 examination. Yoda also wants to record which Jedi Master conducted the examination. To ensure fairness, he plans to occasionally generate reports showing how often each Jedi Master fails students in a particular examination. (E.g., if one Master passes most of the trainees when conducting her Light Saber Combat Level 1 exam, while another Master fails half the students in the same exam, this would be a cause of concern for Yoda.)

Draw an ER diagram for the database. **Make sure to label all the relationships.** You should indicate the attributes but you do not need to include surrogate primary keys or foreign keys at this point.



Notes:

- Successful analysis of this problem must recognize that we are dealing with two cases of instance / type relationship. When a trainee is examined, this is an **instance** of a particular **kind** (or type) of examination. For example, suppose that on January 1 Anakin Skywalker takes Light Saber Combat Level 1 exam administered by Master Yoda and fails. On the same day Obi-Wan Kenobi takes Light

Saber Combat Level 1 exam from Master Qui-Gonn Jinn and passes. A week later Anakin takes Light Saber Combat Level 1 exam from Master Qui-Gonn Jinn and also passes. What we have here is 3 instances of "Light Saber Combat Level 1" examination. Note that it is the **instance** that is associated with a particular student and a particular master, and it is **the instance** for which we record the result (pass or fail). It would be incorrect to associate those things to the type. If we did, we would be able to record that Anakin and Obi-Wan have both taken "Light Saber Combat Level 1" and that Yoda and Qui-Gonn Jinn both administered this kind of exam, but we would not know from whom Anakin took the exam. (We also would not have any place to record when Anakin took each exam and whether he passed or fail.) This is our first case of instance/type distinction.

- In a similar vein, "Light Saber Combat Level 1", "Light Saber Combat Level 2", and "Light Saber Combat Level 3" are all instances of "Light Saber Combat" exams. We could pick different names for the entities, but what is important is that there is a distinction. In my diagram I am using the term "exam_instance" for the specific taking of the exam, "exam_kind" for a type of exam offered at a particular level (e.g., "Light Saber Combat Level 1") and "subject" for the subject topic that is instantiated by each level (e.g., "Light Saber Combat" is a "subject"). Again, what matters here is not the specific labels but the three-way distinction.

- One examination being a prerequisite for another is a recursive relationship on exam_kind. For example, a student must pass "Light Saber Combat Level 2" before taking "Jedi Mind Tricks Level 1" — not just "Light Saber Combat." Passing "Light Saber Combat Level 1" would not be sufficient, because it's not the same exam, even though it is on the same subject.

- There is no reason to represent the reports that are going to be generated by Yoda. Those reports are not part of our data. They are something that we will *derive* from our data. In other words, there are no facts that we need to store about those reports.

- There is no reason to represent the number of times that each student have taken an examination. We can get this number by simply counting the instances of a particular examination that are associated with this student.

- On the most basic level, the boxes that represent the entities must list attributes, not **values**. If your diagram has boxes with lists such as "1", "2", "3" or "Light Saber Combat" and "Linear Algebra", then it's not a meaningful ER diagram.

This was a difficult quiz, so I am keeping this in mind while grading it.

b) Write a CREATE statement for a table called **order**, which will store the following information: a free-form description of the order, the date when the order was placed, the amount charged to the customer, and a foreign key to the table **customer**. Assume that the primary key of **customer** consists of a single integer column called "customer_id." The order table also needs a primary key. (Write your answer on the other side of the sheet.)

```
create table `order` (
 order_id integer not null,
 customer_id integer not null,
 date_placed date,
 description text,
 amount_charged decimal(8,2),
 primary key (order_id),
 foreign key (customer_id)
  references customer (customer_id)
);
```

Notes:

- The name of the table must be escaped (`order`), since “order” is an SQL keyword (as in “order by”). It no a big deal if you missed this, but keep things like this in mind.
- Some notes on data types:
    - The only appropriate types for “date_ordered” are “date” or “datetime.” There is no justification for using “integer” here.
    - Description should be “text” or varchar with a large limit, not varchar(100).
    - Amount charged should be “decimal”, as we want to allow for cents.
- order_id must be declared as a primary key.
- There needs to be a “foreign key” clause.
- customer_id field also needs to be defined in this table. It’s not sufficient to refer to it in the “foreign key” clause. The “foreign key” clause just declares a constraint, it does not define any fields.
- customer_id should probably be declared as “not null,” though, strictly speaking, nothing in the question says that it must be.